

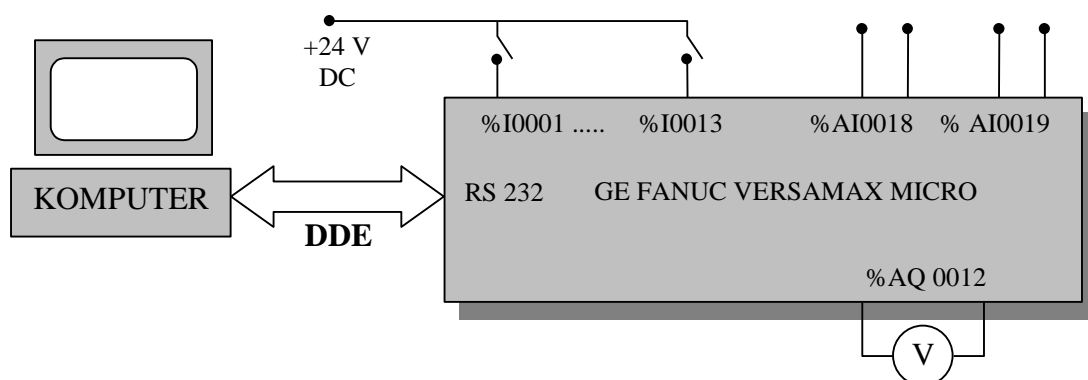
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie KATEDRA AUTOMATYKI <b>LABORATORIUM</b> Aparatura Automatykacji			
<b>Ćwiczenie 3.</b> <b>Sterownik PLC – realizacja algorytmu PID</b>			
Wydział EAIiE kierunek AiR rok II		Zespół 2	Poniedziałek 14:00
L.P.	Imię i nazwisko	Ocena	Data zaliczenia
1.	<b>Łukasz Bondyra</b>		
2.	<b>Paweł Górka</b>		
3.	<b>Jakub Tutro</b>		
4.	<b>Krzysztof Wesołowski</b>		
Data wykonania ćwiczenia		09.03.2009	Podpis

## Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z budową i podstawami działania sterowników PLC, oraz podstawami ich programowania. Sterowniki te programuje się najczęściej za pomocą języka drabinkowego LD (Ladder Diagram). Powstał on gdy sterowniki PLC wchodziły dopiero na rynek. Był on bardzo intuicyjny dla ówczesnych automatyków, którzy realizowali do tej pory układy automatyki na przekaźnikach. Dzięki temu mogli łatwo przejść z układów przekaźnikowych na nowoczesne sterowniki PLC.

## Opis stanowiska doświadczalnego

Uproszczony schemat stanowiska doświadczalnego przedstawia poniższy schemat:



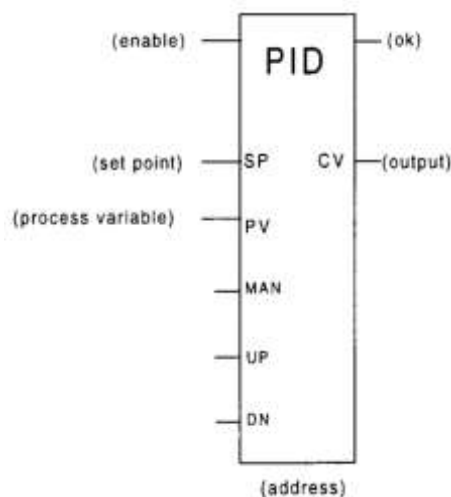
Elementy stanowiska doświadczalnego:

- Sterownik PLC – GE FANUC VERSAMAX MICRO
- Osprzęt do sterownika – przełączniki umożliwiające zadawanie wartości na wejścia binarne sterownika
- Komputer PC z zainstalowanym oprogramowaniem VERSAPRO v1.1 umożliwiającym programowanie sterownika poprzez interfejs RS-232.

## Opis bloku regulatora PID

Regulator PID jest dostępny w programie VERSAPRO jako blok funkcyjny (wartość wyjściowa bloku funkcyjnego – w przeciwieństwie do funkcji – zależy nie tylko od wartości wejściowych, ale także od stanu wewnętrznego bloku). Należy zwrócić uwagę na fakt, że jest to regulator dyskretny, a nie ciągły. Wynika to z samej budowy sterownika, który jest urządzeniem cyfrowym. Blok ten został zaimplementowany w sposób który gwarantuje, że w ciągu jednego cyklu będzie wykonany tylko jedna iteracja algorytmu. Wykorzystuje on 40 kolejnych rejestrów w pamięci. Należy zwrócić uwagę podczas alokowania tych rejestrów, gdyż alokuje się je podając jedynie adres pierwszego rejestru, a dostęp do nich nie jest blokowany i łatwo przez nieuwagę je nadpisać, co spowoduje błędy lub uniemożliwi działanie regulatora. Rejestry te zawierają ustawienia bloku PID jak i wewnętrzne zmienne. Można je zmieniać w programie za pomocą funkcji przesuwania danych takich jak MOVE, lub BLKMOVE, oraz poprzez wpisywanie wartości w okienku „Tuning parameters” dostępnym w menu kontekstowym po kliknięciu prawym przyciskiem myszki na blok regulatora. Szczegółowy opis parametrów i rejestrów pod którymi są zapisywane jest dostępny w instrukcji i nie będę go tutaj przytaczał.

Wygląd bloku przedstawia poniższy rysunek:



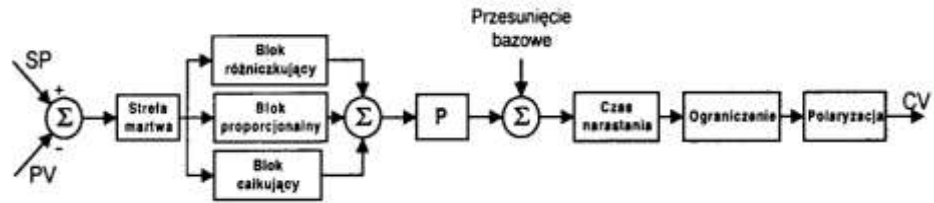
Opis wejść i wyjść:

- **enable** – zezwolenie
- **SP** – set point, czyli wartość zadana
- **PV** – process variable, czyli zmienna procesowa
- **MAN** – przełączanie trybów sterowania ręcznie (1) / automatyczne (0)
- **UP** – zwiększanie wartości wyjściowej w trybie pracy ręcznej
- **DN** – zmniejszanie wartości wyjściowej w trybie pracy ręcznej
- **ok** – potwierdzenie poprawności działania
- **CV** – control variable, czyli zmienna sterująca obliczona przez regulator

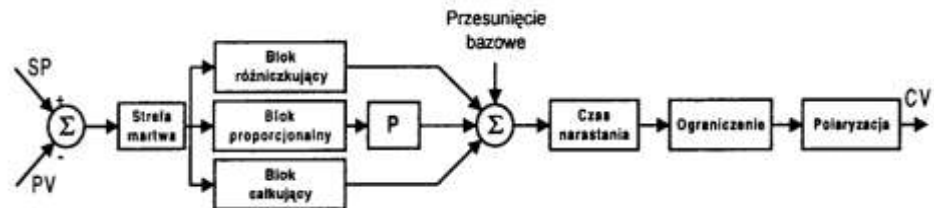
Blok PID istnieje w dwóch wersjach różniących się algorytmem sterowania:

- **PID ISA**  
Realizuje wzmocnienie (mnożenie przez współczynnik P) sygnału stanowiącego sumę sygnałów pochodzących ze wszystkich 3 bloków regulatora: proporcjonalnego, całkującego i różniczkującego.

### Ćwiczenie 3 - Sterownik PLC - realizacja algorytmu PID



- PID IND (Independent Term Algorithm) - przez nas wykorzystywana  
Realizuje wzmocnienie tylko dla części proporcjonalnej regulatora, zgodnie z poniższym schematem:

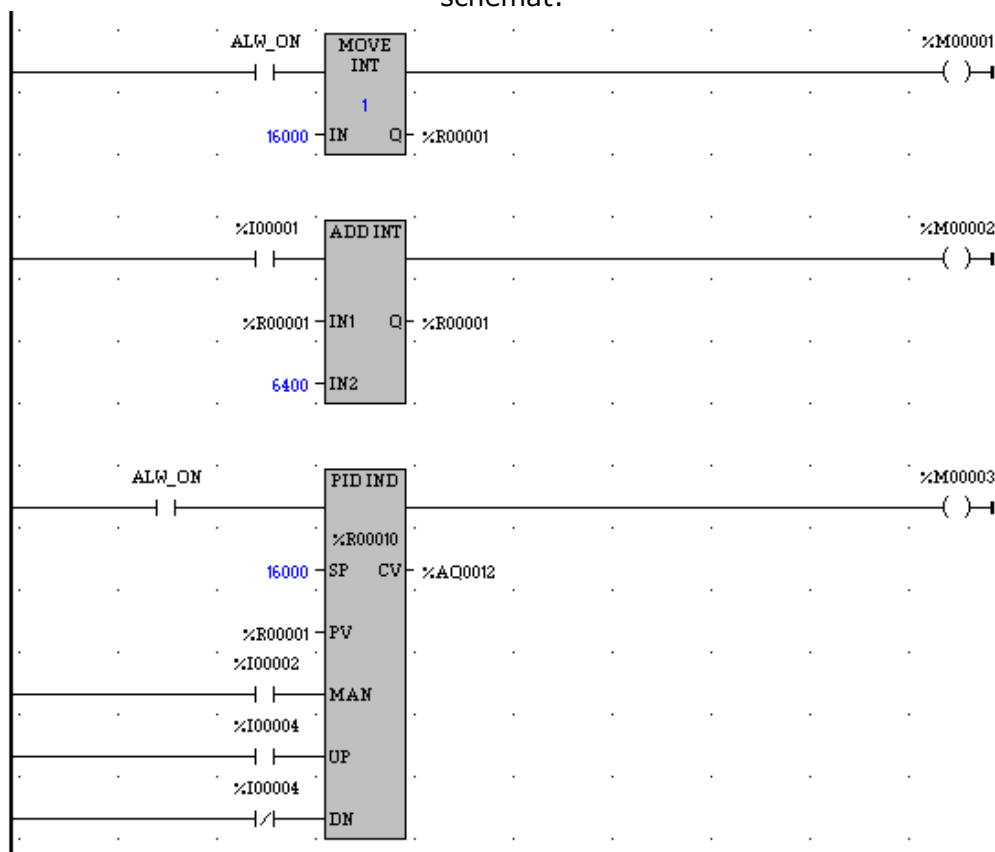


Poza tym obie wersje się niczym nie różnią.

## Wykonanie ćwiczenia

Ćwiczenie rozpoczynamy od ustawienia parametrów połączenia komputera ze sterownikiem. W naszym przypadku należy wybrać port COM1. Kolejnym krokiem jest ustawienie typu sterownika, jakiego będziemy używać, czyli IC200UAL006. Po wybraniu sterownika nie należy zmieniać żadnych parametrów domyślnych. Gdy zakończymy konfigurację możemy przystąpić do programowania.

Głównym zadaniem podczas tego ćwiczenia była realizacja regulatora PID w badanym sterowniku. W tym celu zaprojektowaliśmy w języku drabinkowym następujący schemat:



Pierwszy szczebel realizuje ustawianie wartości zadanej na 50%. Jako, że sterownik GE FANUC VERSAMAX PRO jest pracuje na arytmetyce stałoprzecinkowej przy dwubajtowym słowie, połowa zakresu to liczba 16000. Ta właśnie liczba jest w pierwszym szczeblu wpisywana do rejestru %R00001 za pomocą funkcji MOVE INT. Drugi szczebel realizuje podawanie skoku jednostkowego, czyli dodawanie do wartości początkowej równej stale 50% wartości odpowiadające 20%, czyli 6400. Realizuje to blok ADD INT, którego wejście ENABLE jest sterowane za pomocą wejścia binarnego sterownika %I00001. Umożliwia to włączanie i wyłączenie skoku jednostkowego za pomocą zewnętrznego włącznika, w naszym przypadku to pierwszy włącznik na panelu. W kolejnym szczeblu mamy już regulator PID w wersji PID IND. Regulator ustawiamy jako zawsze włączony, podpinając do jego wejścia ENABLE styk ALW\_ON. Na wejście SP (set point) ustawiamy wartość 50%, czyli 16000, natomiast na wejście PV (process value) wartość rejestru %R00001. Do wejścia ustawiającego tryb pracy – MAN podłączyliśmy wejście binarne sterownika %I00002 odpowiadające drugiemu włącznikowi na panelu. Sterowanie ręczne postanowiliśmy przypisać do włącznika 4 (wejście %I00004). Użyliśmy tego samego włącznika do zwiększania i zmniejszania sterowania podłączając go do styku normalnie zamkniętego i normalnie otwartego.

Po zaprojektowaniu schematu i załadowaniu programu do sterownika przystąpiliśmy do testowania go dla różnych ustawień podanych w instrukcji do ćwiczenia. Wnioski i spostrzeżenia z przeprowadzonych testów przedstawiamy poniżej.

### Obserwacje i wnioski

Jedną z najbardziej znaczących cech regulatora PID realizowanego na sterowniku GE FANUC VERSAMAX PRO jest to, że pracuje on dyskretnie, a nie ciągle. Jest to rzecz jasna podyktowane tym, że sam sterownik jest cyfrowy, a nie analogowy. W związku z tym algorytm sterowania takiego regulatora realizowany jest iteracyjnie co określony interwał czasowy. W przypadku badanego sterownika producent zaleca ustawienie okresu próbkowania na 0,1s. Oznacza to, że regulator próbuje pomiary i oblicza sterowanie 10 razy w ciągu sekundy. Jeżeli patrzymy pod kątem sterowania obiektów, których parametry zmieniają się powoli jest to bardzo duża częstotliwość. Z drugiej strony jeżeli mamy obiekt o bardzo dużej dynamice, to może się okazać, że jest to zbyt rzadko. Trzeba jednak wziąć pod uwagę to, że większość procesów jakimi steruje się w przemyśle, nie ma zbyt wielkiej dynamiki, a te które mają zazwyczaj wymagają już sterowników specjalnych, a nie zwykłych PID.

Kolejnym parametrem, który nie występuje w ciągłych regulatorach analogowych jest strefa martwa. Określa ona jak bardzo musi się zmienić zmienna procesowa, w stosunku do jej wartości w poprzednim cyklu, aby została ona zauważona przez regulator.

Korzystanie przez regulator z wejść na PV i SP zamiast pojedynczego uchybu pozwala nam również zrezygnować z różniczkowania zmian wartości zadanej, dzięki czemu zmniejszamy przesterowanie po zmianie wartości zadanej, nie rezygnując z wpływu części różniczkującej na stabilność. Ma to oczywiście zastosowanie w układach stabilizacji stałej wartości (w układach nadążnych spowodowałoby to pogorszenie czasu dojścia do wartości zadanej)

Realizacja regulatorów PID na sterownikach PLC jest w wielu sytuacjach bardzo dobrym rozwiązaniem. Po pierwsze mając już sterownik PLC możemy na nim dodatkowo zrealizować wiele innych dodatkowych funkcji, albo kilka regulatorów PID. Po drugie możemy także na tym samym sterowniku zrealizować część układów alarmowych i zabezpieczających. Po trzecie cyfrowe wersje regulatorów PID posiadają wiele dodatkowych parametrów, które mogą okazać się bardzo użyteczne. Należy również wziąć pod uwagę korzyści jakie płyną z samego stosowania sterowników PLC, takie jak np. podłączenie sterownika do komputera z systemem SCADA i zarządzanie nim z oddalonej sterowni.